

高职高专计算机精品课程系列规划教材



Java Language Case Course



杨昱曷 主 编
李 锐 副主编
张 伟 主 审

Java语言 案例教程

高职高专计算机精品课程系列规划教材

Java 语言案例教程

杨昱曷 主 编
李 锐 副主编
张 伟 主 审

浙江大學出版社

图书在版编目(CIP)数据

Java 语言案例教程 / 杨昱曷主编. — 杭州: 浙江大学出版社, 2008. 10

(高职高专计算机精品课程系列规划教材)

ISBN 978-7-308-06280-0

I. J… II. 杨… III. ①JAVA 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 157764 号

Java 语言案例教程

杨昱曷 主编

策 划 希 言 冯 骏

责任编辑 黄娟琴

文字编辑 吴昌雷

封面设计 陈 辉

出版发行 浙江大学出版社

(杭州天目山路 148 号 邮政编码 310028)

(E-mail: zupress@mail. hz. zj. cn)

(网址: <http://www.zjupress.com>

<http://www.press.zju.edu.cn>)

电话: 0571-88925591, 88273066(传真)

排 版 杭州中大图文设计有限公司

印 刷 浙江中恒世纪印务有限公司

开 本 787mm×1092mm 1/16

印 张 16.25

字 数 406 千

版 印 次 2008 年 10 月第 1 版 2008 年 10 月第 1 次印刷

印 数 0001—3000

书 号 ISBN 978-7-308-06280-0

定 价 25.00 元

版权所有 翻印必究 印装差错 负责调换

浙江大学出版社发行部邮购电话(0571)88925591

高职高专计算机精品课程系列

规划教材专家指导委员会

主 任

温 涛 大连东软信息技术职业学院院长,教授、博士生导师,教育部高等学校高职高专计算机类专业教学指导委员会主任

副主任

俞瑞钊 宁波大红鹰职业技术学院院长,教授、博士生导师,教育部高等学校高职高专计算机类专业教学指导委员会副主任

顾 问

谭浩强 我国著名计算机教育家,全国高等院校计算机基础教育研究会会长

委 员

蒋川群 上海第二工业大学计算机与信息学院院长,教育部高等学校高职高专计算机类专业教学指导委员会副主任

丁桂芝 天津职业大学电子与信息工程学院院长,教授,教育部高等学校高职高专计算机类专业教学指导委员会委员

刘甫迎 成都电子机械高等专科学校计算机工程系主任,教授,教育部高职高专计算机类专业教学指导委员会委员

胡维华 杭州电子科技大学副校长,教授,教育部高职高专计算机类专业教学指导委员会委员

武马群 北京信息职业技术学院院长,教授,教育部高职高专计算机类专业教学指导委员会委员

张 伟 大连东软信息技术职业学院常务副院长,博士,教育部高职高专计算机类专业教学指导委员会委员

序

高等职业教育作为高等教育发展中的一个类型,肩负着培养面向生产、建设、服务和管理第一线需要的高级职业技术型人才的使命,在我国加快推进社会主义现代化建设进程中具有不可替代的作用。经过数年的探索和实践,我国的高等职业教育已为现代化建设培养了一批高素质的技能型专门人才,对高等教育大众化作出了重要贡献;也丰富了高等教育体系结构,形成了高等职业教育的体系框架,顺应了国民经济各部门、企事业单位对应用型和技能型人才的不同需求。

精品课程是高等职业教育课程建设的重要组成部分,也是高等职业教育教学质量与教学改革示范。浙江大学出版社在省级精品课程和国家“十一五”规划教材课程基础上组织出版的“高职高专计算机精品课程系列规划教材”,是由在高职高专教学第一线有丰富教学经验的教师编写的。整套教材从选题到内容的组织,都着力贯彻了实用性的原则;明确提出了与行业接轨,以就业为导向的编写要求;强调从计算机应用需求出发,构造适应技能型人才培养的教学内容体系,强调理论教学与实验实训密切结合,尤其突出实践体系与技术应用能力的实训环节。教材编写力求内容新颖、结构合理、概念清楚、实用性强,语言通俗易懂、前后相关课程有较好的衔接。据悉,浙江大学出版社还将在此基础上,陆续征集出版后续教材,力争在3到5年内完成一套完整的高职高专计算机专业教材,以满足高职院校计算机教育发展的需求。

本系列教材主要面向高职高专院校,同时也适用于同等学历的职业教育和继续教育。我们希望,通过本系列教材的编写和推广应用,对交流和提高高职院校计算机专业教学的整体水平,促进高等职业技术教育课程体系和教学培训方法的改革,完善高职高专精品课程建设带来新的活力。

温 涛

温涛 大连东软信息学院院长、教育部高等学校高职高专计算机类专业教学指导委员会主任

前 言

互联网是“20 世纪最伟大的发明之一”,它对人类活动产生了深远的影响;21 世纪更是“信息时代”,互联网的应用将会更加广泛而深入。作为一开始就为支持网络应用程序而设计的 Java 语言,具备了与生俱来的网络特性、安全性和跨平台性,以及简单易学性,日益成为互联网开发的主导语言,而且逐步向各个应用领域发展,如在通信和嵌入式应用等领域。Java 是一种应用面广、知识面宽的语言,从 Java 核心编程到 J2EE 企业级应用开发,再到通信和嵌入式编程。作为一个初学者,学好 Java 核心编程是很有必要的,它将为学习 J2EE 企业级应用开发打下一个良好的基础。鉴于此目的,作者在多年的教学实践基础上,结合课程性质和特点,对知识点进行了一定的整合,采用新颖的有利于学生学习和掌握的案例教学模式,编写了此教材。

本书共分 9 章 4 个部分,外加一个附录。第一部分:Java 语言基础,第 1~2 章;第二部分:面向对象编程基础,第 3 章;第三部分:Java 实用技术(Applet、GUI 编程、数据库技术、多线程编程和流处理等)第 4~8 章;第四部分:Java 网络编程基础,第 9 章;附录:Java 常用工具。

本书有如下几个特点:

(1)内容丰富、全面,涵盖绝大部分 Java 编程基础知识,避免了目前部分高职教材存在的简单、粗糙现象。

(2)知识新颖,采用最新的 JDK1.6 API 编程,图形界面采用目前流行的 Swing 组件技术。

(3)语言精练、通顺,图文并茂,案例通俗化、典型、丰富,注释详尽,理论与实践结合紧密,习题量大,适合教学与自学。

(4)结合模块化教学思想,对 Java 网络编程基础和附录,可根据教学需要适当选取教学内容,另外,对 Java 实用技术模块,也可以根据需要进行有选择地进行教学。

(5)采用新颖的案例教学模式展开,每一章节都采用:提出“学习目标”→了解“案例内容”→深入“知识链接”→回顾“总结”→提高“练习题”的教材组织形式,能充分调动学生的学习兴趣。

(6)案例选择精巧,具有围绕知识点的典型性。为了达到实用实效的目的,从第 5 章 Java GUI 编程到第 7 章 Java 数据库技术,主要围绕一个综合性的案例“学生信息管理系统”展开,帮助学生逐步掌握使用 Java 语言解决实际问题的能力。

本书特别适用于高职院校计算机及相关专业的 Java 语言基础课程,同时也可作为编程爱好者的自学教材。

本书由杨昱曷任主编,李锐任副主编,杨昱曷编写了第 1~5 章和第 7 章,周巧婷编写了第 6 章和附录,李锐编写了第 8 章和第 9 章。由于作者水平和经验有限,不足之处在所难免,希望广大读者批评指正。作者 Email:yybljf@163.com。

作 者

2007 年 10 月

目 录

第 1 章 Java 简介与开发环境	1
1.1 我的第一个 Java 程序	1
案例 1 在屏幕上输出“我的第一个 Java 程序”	2
1.2 Java 开发环境	7
案例 2 安装 JDK 和配置环境变量	7
案例 3 JcreatorPro4.5 的安装和创建第一个 Java 应用程序	11
第 2 章 Java 语言基础	17
2.1 Java 语言的基本组成	17
案例 4 求长方形面积	17
2.2 基本数据类型	20
案例 5 数据类型的定义与处理	21
案例 6 不同类型数据间的混合运算	22
2.3 运算符	27
案例 7 利用算术运算符进行算术运算	27
案例 8 利用关系运算符进行判断	28
案例 9 逻辑运算符和关系运算符联合使用	29
案例 10 利用位运算符运算	29
2.4 流程控制	35
案例 11 求星期几	35
案例 12 求阶乘	40
案例 13 求 10~100 之间的素数	44
2.5 数组	48
案例 14 找最大值和最小值	48
第 3 章 面向对象编程基础	53
3.1 类与对象	54
案例 15 自行车品牌	54
3.2 继承	58
案例 16 企业员工信息处理	58

3.3	多态.....	63
	案例 17 利用方法覆盖处理动物叫声	64
	案例 18 利用方法重载求两个数的和	65
3.4	包.....	68
	案例 19 通过包计算面积	68
3.5	修饰符.....	71
	案例 20 通过访问修饰符定义并输出变量	72
	案例 21 利用静态修饰符求赛车数量	75
	案例 22 利用抽象修饰符求员工薪水	77
3.6	接口.....	81
	案例 23 利用接口声明战斗机类和坦克类	81
3.7	异常处理.....	85
	案例 24 被零除异常	85
	案例 25 抛出异常	86
第 4 章 Applet 入门		92
4.1	Applet 基础	92
	案例 26 在屏幕上显示“你好,我是 Applet 小程序”	92
4.2	Applet 的生命周期	96
	案例 27 Applet 生命周期	96
4.3	向 Applet 传递参数.....	100
	案例 28 向 Applet 传递参数	100
4.4	多媒体	102
	案例 29 在 Applet 中显示“蘑菇”图像	102
	案例 30 在 Applet 中播放“红河谷”音乐	104
第 5 章 Java GUI 编程		107
5.1	容器	108
	案例 31 用户登录界面的设计	108
5.2	常用组件	122
	案例 32 学生信息录入界面的设计	122
	案例 33 带有菜单的学生信息管理系统主界面	130
5.3	事件模型	135
	案例 34 实现用户登录界面的功能	135
	案例 35 实现学生信息录入界面的功能	137
	案例 36 带弹出式菜单的主界面	141

第 6 章 多线程	151
6.1 线程的创建	151
案例 37 通过线程实现在主界面显示日期时间	152
6.2 线程的状态及调度	159
案例 38 利用线程调度控制线程的执行	159
6.3 线程同步与通信	163
案例 39 两个线程同往一账户存款问题	163
案例 40 线程通信实现生产者—消费者模型	166
第 7 章 Java 数据库技术	173
7.1 访问数据库	173
案例 41 结合数据库访问的用户登录	174
7.2 带参数的数据库访问	183
案例 42 使用预编译类实现学生信息录入界面功能	183
第 8 章 输入输出流	189
8.1 Java 输入/输出流	189
案例 43 基于字节流的输入输出	190
案例 44 基于字符流的输入输出	191
8.2 从一个流构造另一个流	199
案例 45 过滤器流类演示程序	199
8.3 File 类	204
案例 46 显示当前目录中的文件及目录信息	204
案例 47 显示指定的文件属性	205
第 9 章 Java 网络编程基础	209
9.1 面向连接通信的实现	209
案例 48 面向连接的客户/服务器通信	210
9.2 无连接通信的实现	217
案例 49 面向无连接的客户/服务器通信	217
9.3 利用 URL 访问网站	221
案例 50 访问清华大学网站	222
附录 Java 常用工具	226
1.1 Java 的类库	226
1.1.1 语言基础类库	226
1.1.2 常用包装类的用法	226
1.1.3 基本数据类型之间的转换	232

1.1.4	Object 类	233
1.2	数组与向量类	234
1.2.1	数组	234
1.2.2	向量类	235
1.3	字符串类	238
1.3.1	字符串类的构造方法	238
1.3.2	String 类的应用	239
1.3.3	StringBuffer 类的应用	241
1.4	其他常用工具	243
1.4.1	数学函数类 Math	243
1.4.2	日期类	244
1.4.3	随机数类 Random	246

Java 简介与开发环境

【本章目标】

- ⇒ 了解 Java 语言的特点。
- ⇒ 了解 Java 虚拟机和垃圾回收机制。
- ⇒ 掌握 Java 程序最基本的操作过程和程序结构。
- ⇒ 掌握 JDK 的安装和配置。
- ⇒ 初步掌握 Jcreatorpro4.5 的安装和基本用法。

Java 语言自 1996 年正式发布以来,经历了出生、成长和壮大的阶段,现在已经成为 IT 领域的主流编程语言。Java 起源于 Sun 公司的一个叫“Green”的项目,目的是开发嵌入家用电器的分布式软件系统,使电器更加智能化。当时不叫 Java,而叫 Oak。随着 Internet 的发展,Sun 公司将目标市场转向 Internet 应用程序,并将工程名称修改为 Java。

Java 基于 C 和 C++ 语言构建,它的语法起源于 C 语言,而面向对象的特点是受 C++ 语言的影响。Java 保留了许多 C++ 语言的语法,但去除了明确的资源引用、指针算法与操作符重载等具有潜在危险的特性。

Java 语言具有与硬件平台无关的特性,它在企业或中间件应用程序(如在线商店、事务处理、数据库接口等)以及跨平台应用软件开发方面是一种很流行的语言;另外,在手机和 PDA 等小平台上的应用也很普及。本章主要介绍 Java 语言的特点、基本程序结构、JDK 运行环境的安装和 IDE(集成开发环境)的初步使用。

1.1 我的第一个 Java 程序

Java 是一种优秀的程序设计语言,它最大的优点就是与操作系统无关,即在 Windows、Unix、Linux 以及 Mac OS 等多种操作系统上,都可以使用相同的代码。“一次编写,到处运

行”是 Java 的最大特点,也是 Java 在互联网上被广泛采用的原因所在。



案例内容

案例 1 在屏幕上输出“我的第一个 Java 程序”

下面是一个简单的 Java 程序的案例,我们也可以在记事本中进行编辑,输入如下代码:

```
import java.io.*; //将 java.io 包中的类引入到本程序
public class MyFirstJava{
    public static void main(String args[]){
        System.out.println("我的第一个 Java 程序"); //将我的第一个 Java 程序输出到屏幕上
    } //main 函数结束
} //类结束
```

将此文件保存为 MyFirstJava.java。为方便起见,本例就存在 D 盘根目录下。

进入 DOS 命令状态,在命令行提示符下输入如下命令:

```
d:
javac MyFirstJava.java
java MyFirstJava
```

其中,javac MyFirstJava.java 命令是将 Java 源程序 MyFirstJava.java 编译产生一个 Java 字节码文件 MyFirstJava.class。类文件如果没有指明存放路径,则默认存放在源文件所在路径下,本例为 D 盘根目录下。

java MyFirstJava 命令是用来解释运行编译生成的类文件 MyFirstJava.class。

将此程序编译、运行后,便可在屏幕上显示程序运行结果,如图 1.1 所示。

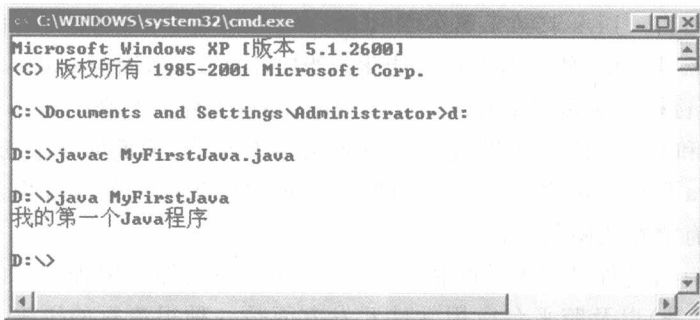


图 1.1 程序 MyFirstJava 的运行结果



知识链接

每一种编程语言都有自己特定的书写格式,Java 也不例外。在这个案例中,我们可以看到 Java 语言有如下特点:

1. Java 程序的组成

Java 源程序主要由 5 部分组成:package 和 import 语句、类、方法和语句。

(1) package 语句用来定义该程序所属的包,该语句必须位于程序的最前面,且每个程序只允许使用一条 package 语句。package 语句在程序中可以省略(如案例 1)。如果省略该语句,则程序属于默认包。关于包的具体概念和作用我们将在第 3 章中介绍。

(2) import 语句用来引入其他包中的类,以便于在程序中使用。该语句必须位于类定义之前,并且可以多次使用,引入多个类。不过,因为 System 是个常用类,该类可以直接使用,也就是说案例 1 的第 1 个语句在这里可以省略。

(3) 类(class)是整个源程序的核心部分,也是编写程序的地方。一个源程序文件至少要有个类。每个类的内容是用一对大括号括起来的,有不同的名字,但是程序的文件名必须和程序主类的名字相同。主类是指 main()方法所在的类。class 是类的定义字,其后是类的名称,public 表示此类是公开的,其他程序也可以调用。公共类的定义格式为:

```
public class 类名
```

```
{
```

```
    语句体
```

```
}
```

关于类的定义与用法详见第 3 章。

(4) 方法:每个 Java 应用程序都要有且只有一个 main()方法,它是程序运行的开始点。main()方法的格式永远都是 public static void main(String args[]),不能随意更改。一个类中可以有多个方法,每个方法都有不同的名字,其定义格式为:

```
修饰符 返回值类型 方法名(参数)
```

```
{
```

```
    语句体
```

```
}
```

在方法的内部不可以再定义其他的方法,但是可以调用其他方法。

(5) 语句:类或者方法中的语句体是由一条条以分号结尾的语句组成的。语句是 Java 程序的基本单位之一,是程序具体操作的内容。每条语句各占一行,以分号结尾。语句有赋值语句、调用方法语句和对象定义语句等多种形式,我们今后将一一讲解。

Java 语言是严格区分大小写的语言,所以在书写语句时,一定要注意大小写不能混淆。例如:在案例 1 所示的程序中,“System.out.println(“我的第一个 Java 程序”);”不可以写成“system.Out.println(“我的第一个 Java 程序”);”。因为在 Java 编译器看来,这是两个完全不同的语句。

“System.out.println(“我的第一个 Java 程序”);”语句是在屏幕上输出“我的第一个 Java 程序”这段文字,并添加一个换行符,使当前光标跳转到下一行。

“//”符号后面的内容为程序的注释部分,它对程序的运行及运行结果不起作用。

2. 应用程序的流程

本案例中,应用程序的运行流程如图 1.2 所示。

3. Java 语言的特点

Java 语言主要有如下几个基本特点:面向对象、简单易学、网络特性、安全性、平台无关性、多线程。

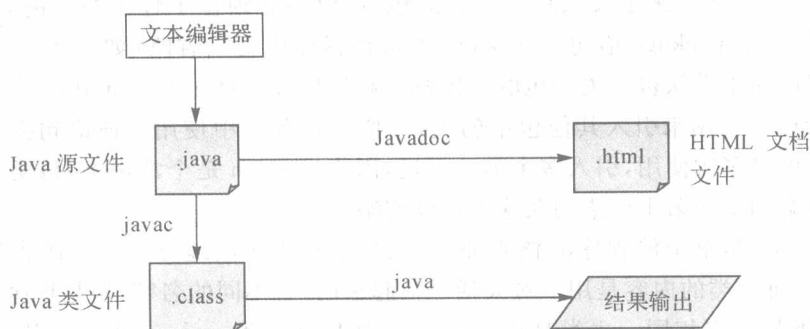


图 1.2 应用程序流程图

(1)面向对象。在 Java 中,任何东西都是对象,它没有独立于类外的函数或变量存在。相对 Java,C++ 就没有那么纯。为了与 C 有一定的兼容性,在 C++ 中,经常可看到一些独立于类外的函数和变量存在,如:main 函数、全局变量的定义等。

(2)简单易学。Java 的设计目的是让程序员觉得既简单易学又好用。假设程序员有过编程经历尤其是已经理解面向对象编程的基本概念,那么学习 Java 将是轻而易举的事。而且,为了使 Java 简单且易于掌握,Java 设计人员去掉了 C 和 C++ 语言中的许多复杂又容易混淆的概念,比如 Java 中不再存在指针、运算符重载和多重继承等功能;Java 无 goto 语句,也无头文件;struct 和 union 等结构也已从 Java 中删除。

(3)网络特性。Java 一开始就是为支持网络应用程序而设计的,因此 Java 作为开发工具在 Internet 等具有不同平台的环境中得到了广泛应用。Java 具备强大且易于使用的连网能力。任何人只要曾经使用另一种语言进行 Internet 编程,就会发现用 Java 来完成一些复杂的任务时,比如打开一个 Socket 连接,整个过程是非常容易的。另外在 Java 中使用了 Servlet(服务器小应用程序)的机制,使得在用 Java 进行服务器端编程时,可获得非常高的效率。许多流行的 Web 服务器都支持 Servlet 远程方法调用机制来实现分布式对象之间的通信。

(4)安全性。由于 Java 应用于网络程序的开发,因而安全性变得至关重要。Java 主要采用了以下几种安全保障机制。

① 采用字节码校验器(Byte Code Verifier)防止 Java 运行环境被装入有害的类文件。

② 采用类装载器(Class Loader)将本机类与网络资源类的名称分开,以保持安全性。

③ Java 的内存分配模式可以有效防止有害代码的入侵。由于取消了指针,程序无法访问它不应该访问的内存,并且 Java 的编译器不处理内存布局,从而无法从类的结构推知运行过程中的实际内存布局。

(5)平台无关性。Java 语言具有最完美的平台无关性。Java 的平台无关是指用 Java 编写的应用程序不用修改就可可在不同的软、硬件平台上运行。一般,平台无关分为源代码级和目标代码级两种,像 C、C++ 等语言虽然也具有一定程度的源代码级平台无关性(即与硬件平台无关),但编译好的目标代码则充分依赖所运行的软件平台(即操作系统),所以对像 C、C++ 等语言编写好的程序在 Windows 操作系统下编译好的目标代码是不可能 Linux 操作系统下直接运行的,必须要重新编译目标代码。Java 则通过使用与平台无关的代码——字节码(byte code),而不是通常的特定机器上的机器码,依靠 Java 虚拟机(JVM)在

目标代码级实现平台无关性。

另外,Java 采用了基于 IEEE 标准的数据类型,即 Java 的原始数据类型在任何机器上都是一样的。例如,整型总是 32 位,长整型总是 64 位等,这样保证了程序可以在任何平台上工作。并且,Java 还提供了一个用于访问底层操作系统功能的可扩展类库,使 Java 的应用程序不依赖于具体的系统,确保了 Java 的平台无关性。

(6)多线程。对多线程的内置支持为程序员提供了提高图形应用程序交互式性能的强有力的工具。如果应用程序需要在滚动页面和从服务器下载文本文件的同时运行动画或播放音乐,则多线程的优点可以提供这种更好的交互性以及实时行为。多线程是在单个进程空间获得快速、并发的方式,所以线程有时也称为轻量级进程。

Java 类库提供了 Thread 类,此类支持启动线程、运行线程、停止线程和检查线程状态等方法。

4. Java 虚拟机(JVM)

JVM 是一种抽象机器,它附着在具体操作系统之上,本身具有一套虚拟的机器指令,并有自己的栈、寄存器组等,它是 Java 平台无关的基础。但是,JVM 通常是在软件上而不是在硬件上实现。在 JVM 上,由 Java 解释器来解释 Java 编译器编译后的程序,即 Java 编程人员在编写软件之后,通过 Java 编译器将 Java 源程序编译为 JVM 的字节码。

任何一台机器,只要配备了 Java 解释器就可以运行这个程序,从而使 Java 程序可以方便地被移植到网络上的不同机器上。图 1.3 为一个 Java 程序的执行过程。

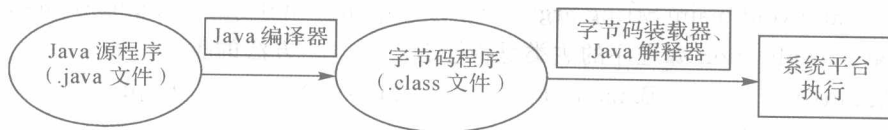


图 1.3 Java 程序的执行过程

5. 垃圾回收机制

用 C 及 C++ 编写软件的编程人员必须仔细跟踪忙碌的内存块。当一个块不再使用时,必须让程序释放它才能再用。在大的项目中,这可能很困难,并往往成为出错和内存不足的根源。

在 Java 语言的情况下,编程人员不必为内存管理操心。Java 语言系统有一个叫做“无用单元收集器”的内置程序,它扫描内存,并自动释放那些不再使用的内存块。

Java 语言的这种自动垃圾回收机制,对不再被程序引用的对象将自动取消其所占资源,这就彻底消除了内存泄漏之类的错误,并免去了程序员管理存储器的繁琐工作。

总结

◇ Java 编程的操作过程:

(1) 可以使用任意一种文件编辑工具输入程序代码。

(2) 用 javac 命令编译源程序。

(3) 编译通过后,用 java 命令执行字节码程序。

◇ Java 语言的特点:

面向对象、简单易学、网络特性、安全性、平台无关性、多线程。

◇ Java 程序的基本结构:

Java 源程序主要由 5 部分组成:package 语句、import 语句、类、方法和语句。



练习题

1. 选择题

(1) Java 语言具有下列特点:_____。

- A. 面向对象 B. 平台无关 C. 安全性 D. 以上都正确

(2) Java 源代码程序的扩展名为_____。

- A. .class B. .java C. .html D. .exe

(3) Java 编译器会将 Java 源程序转换为_____。

- A. 字节码 B. 可执行代码 C. 机器代码 D. ASCII 码

(4) main() 方法是 Java Application 程序执行的入口点,对于 main() 方法的方法头,以下合法的项是_____。

- A. public static void main() B. public void main(String args[])
C. static void main(String args[]) D. public static void main(String args[])

(5) Java Application 源程序的主类是指包含_____方法的类。

- A. init() B. main() C. paint() D. println()

2. 简答题

(1) 简要描述 Java 语言的特点。

(2) 简要描述以下关键字的含义和作用:public、class、static、void、String。

3. 看程序写结果

```
import java.io.System;  
public class MyFirstApp{  
    public static void main(String args[ ]){  
        String temp = "我的";  
        temp = temp + "第一个";  
        System.out.println(temp + "Java 程序!");  
    }  
}
```

4. 编程

编写一个 Java 程序,要求在屏幕上输出如下文字:

您好,我是一名大学生。

我正在努力学习 Java 语言...

1.2 Java 开发环境

开发 Java 程序需要具备如下环境：

- * JDK(Java 开发工具包),是必须的。
- * IDE(Java 集成开发环境),快速有效地开发时需要用到。

JDK 是 Java 开发工具包。它包括:类库、编译器、调试器、Java 解释器、Java 小程序浏览器等,有了这个工具包,就可以在 Java 标准平台(J2SE)上进行 Java 程序的开发。



案例内容

案例 2 安装 JDK 和配置环境变量

JDK 是由 Sun 公司开发并免费提供的 Java 语言软件开发工具包(Java Development Kit, JDK),可通过 Internet,从 Sun 公司的网站直接下载安装。另外,JDK 也被一些 Java 集成开发环境(IDE)所包含,如由 Borland 公司开发的 JBuilder、BEA 公司开发的 Weblogic、IBM 公司开发的 WebSphere 等。

1. JDK1.6 的下载安装:

(1)点击下面网址进入 JDK1.6 下载页面:

<http://java.sun.com/javase/downloads/index.jsp>

(2)点击“Get the JDK download”下载 JDK,如图 1.4 所示。

Java SE Downloads



It's time

Download the complete environment and runtime environment

» Get the JDK download

→ 点击

图 1.4 点击“Get the JDK download”

(3)选择“Accept”,如图 1.5 所示。